Course Title: **Compilers and Languages** Course Code: CCE3113 3rd year
Date: 28.11.2015 (First term) Allowed time: 1 hrs and 30 minutes

Answer the following questions:

Question No. 1 **(6 marks)**

1. Given the following ambiguous context free grammar **(4 marks)**

S ⟶ aSbS | aS | c

(a) Show that the string **s = aacbc** has two leftmost derivations.

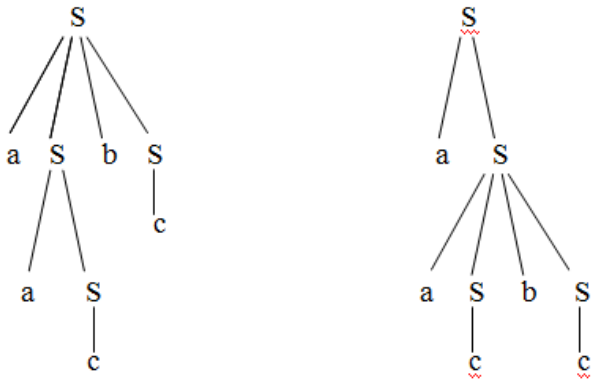
The string aacbc has the following two distinct leftmost derivations:

S → aS ⇒ aaSbS ⇒ aacbS ⇒ aacbc

S → aSbS ⇒ aaSbS ⇒ aacbS ⇒ aacbc

(b) Show the two derivation trees for the string **s**.

The string also has two distinct parse trees, corresponding to the two leftmost derivations.



(c) Find an equivalent unambiguous context-free grammar.

S ⟶ aSA | c

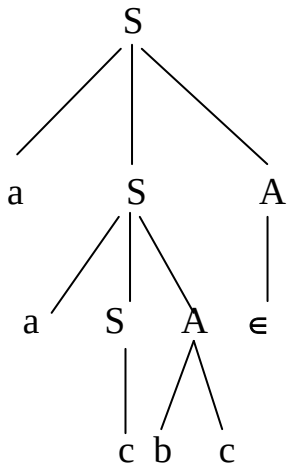
A ⟶ bS | ∈

(d) Give the unique leftmost derivation and derivation tree for the string **s** generated from your rewritten unambiguous grammar.

The unique leftmost derivation

S → aSA ⇒ aaSAA ⇒ aacAA ⇒ aacbSA⇒ aacbcA⇒ aacbc

The derivation tree



2. Consider the following grammar with terminals T = {a, b}. **(2 marks)**

$$S \longrightarrow S\ a \mid b$$

(a) Can this grammar be recognized by a recursive descent parser? Why or why not?

Recursive descent does not work with this grammar because this is a **left-recursive grammar** has a non-terminal S where $S \rightarrow^{+} S\alpha$ for some α

(b) If not, How can you rewrite this grammar to make it a recursive descent grammar

To make this grammar a recursive descent grammar, this left-recursion must be eliminated

$$S \rightarrow b\ S'$$
$$S' \rightarrow a\ S' \mid \varepsilon$$

Question No. 2 **(5 marks)**

1. Draw a NFA for the regular expression **(2 marks)**

((b|a)+c)+

Answer by yourself

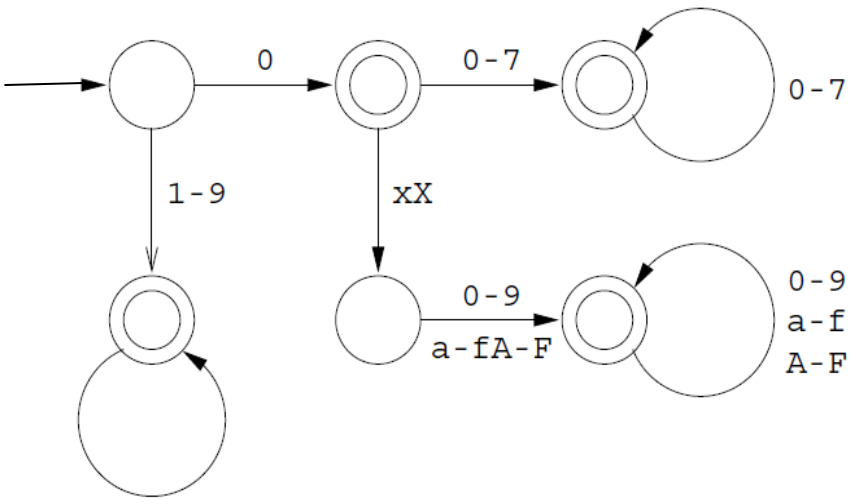
2. C integer literals are formed in the following way: **(3 marks)**

- Octal literals consist of a leading 0 followed by zero or more octal digits (0 through 7)
- Hexadecimal literals consist of a leading 0 followed by x or X followed by one or more hexadecimal digits ((0 through 9, a through f, or A through F)
- Decimal literals consist of one non-zero decimal digit (1 through 9) followed by zero or more decimal digits (0 through 9).

(a) Give a regular expression for C integer literals.

$$0(0-7)^* \mid 0(x \mid X)(0-9 \mid a-f \mid A-F)^+ \mid (1-9)(0-9)^*$$

(b) Draw the state diagram of a DFA (not an NFA!) for this literal form.



Question No. 3 **(9 marks- *each one is worth1 points*)**

True or False?

1. Lexical analysis is recursive in order to handle nested parentheses. False
2. Scanners don't know anything about the grammar of a language. True
3. A successful parse means the input is semantically correct. False
4. Finite State Machines can have an unlimited number of states. False
5. A regular expression is a type of pattern used to classify lexemes. True
6. You can change state in a DFA without reading any input character. False
7. Regular expressions cannot be used to match strings of balanced parentheses. True
8. All Finite State Machines can have only one edge leaving the same state labeled with the same label (character). False
9. A DFA must have exactly one final (accepting) state. False

Best wishes

Dr. Sherin El Gokhy